Profiling Instructor Activities Using Smartwatch Sensors in a Classroom

Zayed Uddin Chowdhury
Computer Science
Georgia Southern University
Statesboro, Georgia, USA
zc01698@georgiasouthern.edu

Pradipta De
Computer Science
Georgia Southern University
Statesboro, Georgia, USA
pde@georgiasouthern.edu

Andrew A. Allen
Computer Science
Georgia Southern University
Statesboro, Georgia, USA
andrewallen@georgiasouthern.edu

ABSTRACT

During a classroom session, an instructor performs several activities, such as writing on the board, speaking to the students, gestures to explain a concept. A record of the time spent in each of these activities could be valuable information for the instructors to virtually observe their own style of instruction. It can help in identifying activities that engage the students more, thereby enhancing teaching effectiveness and efficiency. In this work, we present a preliminary study on profiling multiple activities of an instructor in the classroom using smartwatch sensor data. The proposed approach uses data from available sensors in the smartwatch and builds a machine learning model to predict the activities of an instructor. We use a benchmark dataset that was collected in the wild to test out the feasibility of classifying the activities. Different machine learning models are used and the results are compared using multiple metrics to show the efficacy of predictive modeling in automatic classroom observation of instructors.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Applied computing** → *Education.*

KEYWORDS

classroom observational study, instructor activity recognition, machine learning, smartwatch sensor, multi-label dataset, logistic regression, decision tree, random forest, LSTM

ACM Reference Format:
Zayed Uddin Chowdhury, Pradipta De, and Andrew A. Allen. 2020. Profiling Instructor Activities Using Smartwatch Sensors in a Classroom. In *2020 ACM Southeast Conference (ACMSE 2020), April 2–4, 2020, Tampa, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3374135.3385290

1 INTRODUCTION

Methodology for evaluating instructor's performance is an important topic of classroom observational study, as it has direct effect on students' academic performance. Currently, instructor evaluation is done mainly through student feedback based on a standard survey mechanism known as "Student's Evaluating Teaching (SET)" [16]. However, there is possibility of automating this whole procedure, if we can find correlation between instructor's activity and students' attentiveness. There are already active researches going on for finding student attentiveness [27]. We are focusing on tracking instructor's activities.

Human activity Recognition (HAR) is an emerging field of research in pervasive computing [21]. It has applications in healthcare [13], activity-based crowdsourcing and surveillance [17], targeted advertising [5] etc. Many of the applications of HAR have been surveyed recently [19] [1], from which we can get a summary of the uses and applications of HAR.

Researches on the field of sensor based HAR is gaining momentum with increasing use of smart devices like smart phone and smart watch. These devices contain multiple sensors like accelerometer, gyro sensor, microphone etc. Some other important reasons for the popularity of sensor based activity recognition [2] are their compact size, low-power requirement, low cost, non-intrusiveness in contrary to the previously popular audio and video data based activity recognition techniques [9].

There are some challenges while doing activity recognition. Same activity may be performed by different persons in different manner. Also, the same person can perform the activity in different manner at different time based on the environment, physical or mental condition. This is known as interclass variability. On the other hand, there are similarities between different activities. For example, walking and running has similarities between them. Then there is class imbalance problem. It is a very well known issue in machine learning. A person may do a specific activity rarely comparing with other activities. For example, an instructor may sit down in the classroom for a very little time on contrary to standing up or writing in the board. As a result, it becomes difficult for the machine learning algorithm to classify rare activities. Activities can be performed simultaneously. For example, an instructor may work on computer while he is sitting. This type of machine learning problem is known as multi-label classification [24].

In this paper we have done a preliminary study on classifying human activities based on sensor data of smartwatch. We have used a benchmark dataset [25] which contains multi-label dataset of human activities performed in the wild, that is the dataset was not collected in any experimental setup. We have chosen this dataset because the actual dataset that will be collected in future from instructors will also be multi-label and will be collected in a non-intrusive manner. We have tried to classify 4 activities: walking, sitting, working in computer and standing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
ACMSE 2020, April 2–4, 2020, Tampa, FL, USA
© 2020 Association for Computing Machinery.
ACM ISBN 978-1-4503-7105-6/20/03…$15.00
https://doi.org/10.1145/3374135.3385290

We have tried both traditional approaches like binary relevance [23] using Random Forest [14], Decision Tree [20], Logistic Regression [10] and also neural network based approaches like vanilla neural network [28] and Long Short Term Memory (LSTM) [8] based Recurrent Neural Network (RNN).

2 RELATED WORK

Most of the works related to human activity recognition are based on either wearable sensors, audio or video.

[16] uses motion templates of instructor activities and describes them through a Bag-of-Deep features (BoDF) representation. Deep spatio-temporal features were extracted from motion templates and then utilized to compile a visual vocabulary. After that the visual vocabularies were quantized to optimize the learning model. The activities given below were recognized with an accuracy of 85.41% - Pointing towards the student, pointing towards board or screen, idle, interacting, sitting, walking, using a mobile phone, and using a laptop. For real time action recognition they planned to use explore temporal action segmentation method [15], as instructors perform multiple activities sequentially.

In [4] automatic analysis of teachers' instructional strategies were investigated from audio recordings collected in live classrooms. Dataset was collected from classroom recordings of teachers' audio. Supervised machine learning models were used to train five key instructional segments (Procedures and Directions, Supervised Seatwork, Question and Answer, Small Group Work, and Lecture). The models were validated independently of the teacher to increase the generalizability of new teacher from the same data sample. The five instructional segments above where identified with F1 scores ranging from 0.64 to 0.78. The proposed model were able to detect five instructional segments well above chance level. The system used only low-level features derived only from teachers' audio.

In [18] a proof of concept was designed using a methodology based on the deep learning framework which reduces the difficulty of the optimal feature selection problem significantly. A wrist worn accelerometer was used to identify three basic movements of the human forearm. The validation of the proposed model was done by means of different pre-processing systems and noisy data condition which was assessed using three possible methods. The results showed that the model achieved an average recognition rating of 99.8% which was more than on K-means clustering, linear discriminant analysis and support vector machine. In this paper, comparative analysis between conventional methods like Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), K-means clustering and non-conventional method like Convolutional Neural Network (CNN) were done using different preprocessing steps and training with noisy data. As a result, they found out CNN is very promising in handling the feature engineering process and produces high accuracy if design parameters are defined in an efficient way. Also the proposed model was able to classify daily living activities in real-time and practical scenarios. The paper suggested that the system could be extended towards increasing the number of subjects and also towards people suffering from neurodegenerative diseases.

In [22] 3 motion sensors (accelerometer, gyroscope and linear acceleration sensor) were evaluated at both wrist and pocket positions in order to recognize human activities. Using three classifiers, it was shown that the combination of these two positions outperforms the wrist position alone, mainly at smaller segmentation windows. Since less-repetitive activities, such as smoking, eating, giving a talk, and drinking coffee, cannot be recognized easily at smaller segmentation windows, unlike repetitive activities eg: walking, jogging and biking; 7 window sizes (2–30 s) on thirteen activities were used and how increasing window size affects these various activities in different ways were analyzed. It was found that combining the data from the motion sensors from wrist and pocket positions improves recognition for complex activities and this combination outperforms the wrist only postion's performance in most cases. But the recognition of complex activities is improved with increasing window size. Similar trends were seen for walking and using stairs. However, only increasing the window is not enough for these activities, because the main increase in their recognition performance comes from adding the gyroscope with the accelerometer, either at the wrist or both wrist and pocket positions. Improvements were seen due to increasing window size for simpler activities when their reference performances were low. Though the sensor combinations improved the recognition of complex activities at smaller window sizes, the paper recommended to use a bigger window size for their reliable recognition.

3 METHODOLOGY

In this paper, the dataset [25] we are using has two types of data:

- Feature extracted data
- Raw sensor data

The methodologies used are different for these two types of data.

3.1 Methodology for Feature Extracted Data

3.1.1 Problem Formulation. Let, $D = (X_i, Y_i)$, $1 <= i <= N_d$ which represents the training dataset. Here, N_d = number of training samples, X_i = Features of i^{th} training data, Y_i = labels of i^{th} training data. We define the features as, $x = \{x_1, x_2, x_3, ..., x_{N_f}\}$ which is a set of real values and where N_f = number of features. We define the labels as, $y = \{y_1, y_2, y_3, ..., y_{N_l}\}$, which is a set of binary values and where N_l = number of classes (activities).

For an unseen instance $x = \{x_1, x_2, x_3, ..., x_{N_f}\}$, our target is to build a classifier $h(.)$ which predicts $y = \{y_1, y_2, y_3, ..., y_{N_l}\}$ as a vector of labels for x.

3.1.2 Data Resizing. There are many features and labels in the dataset, but as our context is to detect activities of instructor in a classroom using smartwatch sensors, we selected only the relevant ones. In this paper, we worked only with the accelerometer data. For the aforementioned reasons, the first thing we had to do is to get rid of the rows in the dataset that are not relevant to the selected activities. The second step was to remove rows, which did not have any accelerometer data.

3.1.3 Feature Scaling. Feature scaling or normalization is a very important step in data preprocessing. It is used to normalize the value range of features to a common scale. It is essential when features have different ranges. Otherwise, the model may skew towards specific features only because of it's value range. We have

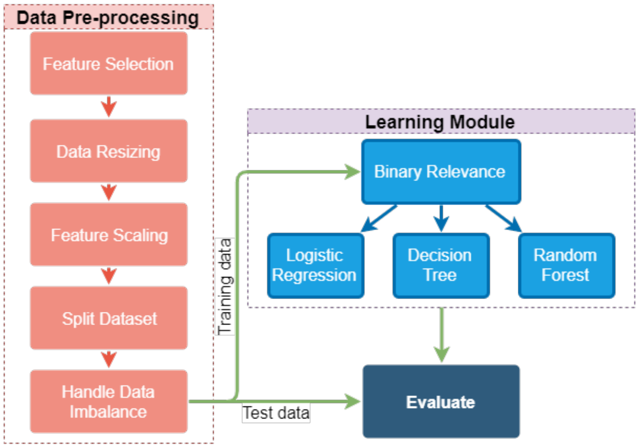

Figure 1: Methodology for Feature Extracted Dataset

used min max scaling method.

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

where x' is the normalized value and x is the actual value.

3.1.4 Dataset Splitting. We primarily split the dataset into training and test dataset with a ratio of 70:30. The training dataset is used for training the models. The models were tested using the test dataset. For the neural network, we also used validation dataset, for tuning hyper parameters. We took 30% of the training dataset as validation dataset.

3.1.5 Handling Data Imbalance. When the instances of one class outnumbers the instances of another class, it is called an imbalanced dataset [6]. The dataset used in this paper is highly imbalanced. Implementing machine learning models using imbalanced dataset is always challenging [12]. There are multiple methods for overcoming the affect of data imbalance [11]. We have used an algorithmic level approach known as Cost-sensitive learning, that is to define fixed and unequal misclassification costs between classes [3]. We adjusted the weights in such a way that it is inversely proportional to class frequencies in the input data.

$$w_i = \frac{n}{k \times n_i}$$

where w_i is the weight to class i, n is the number of observations, n_i is the number of observations in class i and k is the total number of classes. The training settings is summarized in Figure 1. We have used a problem transformation method for multi label binary classification known as Binary Relevance, which essentially considers the prediction of each class as an independent classification problem. After transforming the problem, we tried 3 algorithms for classification and compared them. These 3 estimators are Logistic Regression, Decision Tree and Random Forest.

3.2 Methodology for Raw Data

In this method, we have tried to train a Recurrent Neural Network (RNN) to learn sequence of sensor data.

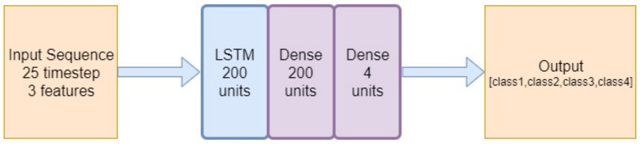

Figure 2: Structure of LSTM Network 1

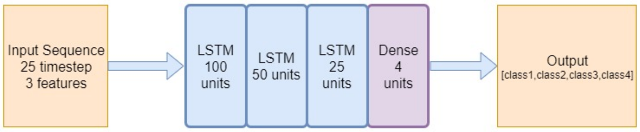

Figure 3: Structure of LSTM Network 2

3.2.1 Problem Formulation. Let, $D = (S_i, Y_i), 1 <= i <= N_d$ which represents the training dataset. Here, N_d = number of training samples, $S_i = i^{th}$ sequence of training data, Y_i = labels of i^{th} sequence. Each sequence S_i is a $N_s \times N_f$ dimensional vector, where N_s = sequence length and N_f = number of features. We define the labels for each sequence as, $y = \{y_1, y_2, y_3, ..., y_{N_l}\}$, which is a set of binary values and where N_l = number of classes (activities).

For an unseen sequence instance $x = \{x_1, x_2, x_3, ..., x_{N_s}\}$, our target is to build a classifier $h(.)$ which predicts $y = \{y_1, y_2, y_3, ..., y_{N_l}\}$ as a vector of labels for x.

3.2.2 Sequence Creation and Labeling. For sequence learning, at first we needed to pre-process the raw data to create sequences. In this paper, sequence size of 25 is used. Raw accelerometer data was recorded in 25Hz frequency. So, each sequence represents 1 seconds of data. We first found out the labels for each user's raw data from the processed dataset's timestamp. Then saved the sequences and labels in a compressed format, so that we can load it easily later. The compression task was important, as the actual raw accelerometer data size was around 10GB.

3.2.3 Dataset splitting. The sequence data is split into training and test set in 70:30 ratio. The training set is used to train the model. 30% of the training data is used for validating the dataset and tune the hyper parameters accordingly. Finally, the test dataset is used to evaluate the model.

3.2.4 Training Settings. LSTM (Long Short Term Memory) based Recurrent Neural Network (RNN) is used for the sequence learning approach. The raw sensor data are multivariate time series data. So, we can describe an activity by a sequence of raw sensor data.

2 LSTM based models are used. One with single LSTM layer and another one using multiple LSTM layers. For both models, the output layer is a fully connected neural network with 4 neurons, each of which outputs 1 if the sample sequence is classified as the corresponding class, otherwise it outputs 0.

Activation Function: As the output is 0 or 1 for each class, sigmoid activation function is used,

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

.

Loss Function: The models are compiled using binary cross entropy loss function, because this is a multi-label classification and we have to treat each output label independently.

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=0}^{N}\{y \log \hat{y} + (1 - y)(\log(1 - \hat{y})\}$$

where $\hat{y}$ is the predicted value, y is the actual value and N is the number of samples. Binary crossentropy measures how far away from the true value (which is either 0 or 1) the prediction is for each of the classes and then averages these class-wise errors to obtain the final loss. However, we also use another version of the same formula which takes class imbalance into account.

The training settings are summarized in Figure 2 and Figure 3.

3.3 Experiments and Results

3.3.1 Dataset. The dataset used here was presented in this paper [25]. It contains over 300k sensor data with context labels from 60 subjects. The data were collected in-the-wild, meaning, it contains data which were recorded using: 1) Naturally used devices, 2) Unconstrained device placement, 3) Natural environment, 4) Natural behavioral content. The dataset contains both smartwatch and smartphone sensors' data. However, we have used only smartwatch accelerometer data for this paper. Tri-axial acceleration from the watch was recorded at 25Hz for around 20sec in each minute. The smartwatch data is divided into 2 parts. One is the processed data and the other is the raw data. The smartwatch accelerometer data contains data from 56 users and the processed dataset contains 210,716 examples. The whole dataset is multi-labeled and contains 51 labels. However, we only took portion of the dataset which corresponds to the 4 activities that we have worked on in this paper. Every entry in the processed dataset was labeled using One Hot Encoding method. For example, label "1010" means the user was doing the 1st and 3rd activities when the data was recorded.

3.3.2 Metrics. We measured different metrics. The most important of them are the F1 score, Area under curve (AUC) and the Balanced Accuracy. There are few reasons behind it. First of all, classification accuracy is a misleading metric in imbalanced dataset. For example, if a dataset contains only 10% positive examples, a classifier that predicts negative all the time will have a high accuracy of 90%, but will be useless. Sensitivity (recall) refers to the individual class accuracy and is calculated as $R = \frac{T_p}{T_p+F_n}$ and specificity $S = \frac{T_n}{T_n+F_p}$ are important measures. F1 score is one of the recommended metrics to be used for unbalanced data.

$$F1 = \frac{2PR}{P + R}$$

where precision $P = \frac{T_p}{T_p+F_p}$. Tp = true positives, Tn = true negatives, Fp = false positives and F_n = false negatives. However, precision and F1 are less fitting in this case, since they are very sensitive to how rare labels are. Balanced accuracy, $BA = 0.5 \times (recall + specificity)$, does not suffer from the aforementioned issues. Area under curve (AUC) is very popular specially in case of binary classification technique [7]. So, as for our case the problem is multi-class and multi-label, we have also calculated average AUC.

Table 1: Results of Processed Dataset (Unweighted)

(a) Linear Regression

Label	BA	AUC	F1
Sit	0.65	0.72	0.55
Walk	0.60	0.84	0.32
CW	0.50	0.69	0.00
Stand	0.50	0.70	0.00
AVG	0.56	0.74	0.22

(b) Decision Tree

Label	BA	AUC	F1
Sit	0.69	0.69	0.63
Walk	0.64	0.64	0.31
CW	0.63	0.63	0.33
Stand	0.60	0.60	0.28
AVG	**0.64**	0.64	0.39

(c) Random Forest

Label	BA	AUC	F1
Sit	0.77	0.86	0.73
Walk	0.63	0.87	0.40
CW	0.57	0.84	0.24
Stand	0.54	0.82	0.15
AVG	0.63	**0.85**	**0.38**

(d) Neural Network

Label	BA	AUC	F1
Sit	0.72	0.80	0.66
Walk	0.60	0.86	0.32
CW	0.51	0.78	0.05
Stand	0.51	0.78	0.02
AVG	0.59	0.81	0.26

Table 2: Results of Raw Dataset (Unweighted)

(a) LSTM 1

Label	BA	AUC	F1
Sit	0.71	0.78	0.66
Walk	0.55	0.83	0.17
CW	0.52	0.76	0.08
Stand	0.52	0.74	0.06
AVG	**0.58**	0.78	0.24

(b) LSTM 2

Label	BA	AUC	F1
Sit	0.71	0.79	0.67
Walk	0.55	0.83	0.2
CW	0.52	0.77	0.09
Stand	0.52	0.74	0.07
AVG	**0.58**	**0.79**	**0.26**

3.3.3 Experiments on Processed Dataset. In Table 1(a), 1(b), 1(c), and 1(d) - F1 score, Balanced Accuracy (BA), Area Under Curve (AUC) is shown as performance metrics for processed dataset without weight adjustment. From the results here, we can see that according to average Balanced Accuracy, the 4 activities are best classified by Decision Tree, but Random forest is also a very close competitor. According to average AUC and average F1 score, all the activities are best classified using Random Forest in the unweighted dataset.

3.3.4 Experiments on Raw Dataset. In Table 2(a), and 2(b) - F1 score, Balanced Accuracy (BA), Area Under Curve (AUC) of Raw dataset without weight adjustment is shown. From the results, we can see that according to all the metrics (Average Balanced Accuracy, Average AUC, Average F1 score) the 2nd multi-layered LSTM model performs slightly better than the first one. Moreover, The cell count in the 1st model is 2 times the cell count in the 2nd model. If we increase the cell count in the 2nd model, it may perform much better. So, we can say that multilayered model performs better than single layered model.

3.3.5 Effect of Class Weights Adjustment. From Table 3(a), 3(b), 3(c) and 3(d), we can see that according to average Balanced Accuracy, Linear regression does the best to classify the activities. According

Table 3: Results of Processed Dataset (Weighted)

(a) Linear Regression

Label	BA	AUC	F1
Sit	0.67	0.72	0.62
Walk	0.76	0.84	0.30
CW	0.64	0.69	0.28
Stand	0.66	0.71	0.30
AVG	**0.66**	0.71	0.30

(b) Decision Tree

Label	BA	AUC	F1
Sit	0.69	0.69	0.63
Walk	0.63	0.63	0.31
CW	0.62	0.62	0.33
Stand	0.60	0.60	0.28
AVG	0.64	0.64	**0.39**

(c) Random Forest

Label	BA	AUC	F1
Sit	0.77	0.86	0.72
Walk	0.61	0.87	0.34
CW	0.57	0.85	0.24
Stand	0.54	0.82	0.16
AVG	0.62	**0.85**	0.37

(d) Neural Network

Label	BA	AUC	F1
Sit	0.73	0.80	0.68
Walk	0.61	0.86	0.34
CW	0.51	0.79	0.04
Stand	0.51	0.76	0.04
AVG	0.59	0.80	0.28

Table 4: Results of Raw Dataset (Weighted)

(a) LSTM 1

Label	BA	AUC	F1
Sit	0.71	0.78	0.68
Walk	0.75	0.84	0.27
CW	0.69	0.76	0.33
Stand	0.68	0.75	0.32
AVG	0.71	0.78	**0.40**

(b) LSTM 2

Label	BA	AUC	F1
Sit	0.72	0.79	0.69
Walk	0.76	0.84	0.27
CW	0.70	0.77	0.33
Stand	0.69	0.76	0.32
AVG	**0.72**	**0.79**	**0.40**

to average AUC, Random Forest does the best. According to average F1 score, Decision Tree performs better. If we take the average of all the scores, Random Forest outperforms the other ones.

From the results in Table 4(a) and 4(b), we can see that for weight adjusted raw dataset, according to the metrics (Average Balanced Accuracy, Average AUC), the 2nd multi-layered LSTM model performs better than the single-layered LSTM model. According to average F1 score, the performances of both the models are similar. However, the 2nd model contains lesser cells than the 1st model. With the increase of cell number, it may perform even better. So, overall we can say that mult-layered LSTM models perform better than the single-layered LSTM models.

Now, comparing same methods we see that, weight adjustment is effecting the processed dataset very slightly. However, it has a great impact on the raw dataset. Specially, the Balanced Accuracy and F1 scores have a positive impact after adjusting the weights. For the 1st LSTM model, the BA increases from .57 to .71, which is around 14% increase. Also, the F1 score increases from .25 to .40 (around 15% increase) for this model. Similarly, for the 2nd LSTM model the BA increases from .58 to .72 (around 14% increase) after weight adjustment. The F1 score goes from 0.26 to 0.40 (around 14% increase) after weight adjustment.

4 DISCUSSION

We generated the activity prediction models based on two classes of techniques - one that requires extensive feature selection and engineering, such as random forest, and the second approach is using Recurrent Neural Network (RNN), where the features are discovered from the raw data. Although Random Forest performs well, we would recommend to use the RNN for the ease of use for this specific problem. The disadvantage of using RNNs is that the raw dataset is an order of magnitude larger than the processed feature set provided as input to generate the Random Forest. Therefore, training a RNN is significantly more costly in terms of resource and time. Note that the dataset has imbalance with respect to sample count per label type. We observed that assigning class weights has a positive impact on the performance of a machine learning model.

In [25], 5-fold performance evaluation (BA) was done in this same dataset. In their paper, the results using only watch accelerometer data for classifying Sitting, Walking, Computer work and Standing are respectively 0.68, 0.75, 0.62, and 0.67. Logistic Regression was used as the classification technique. In [26], Multi Layer Perceptron (MLP) was used with multiple layers on the same dataset. They got an accuracy of 0.75, 0.8, 0.72, 0.63 for the same activities. In our paper, using LSTM we were able to achieve BA of 0.72, 0.75, 0.69, 0.68 for those activities.

This dataset helped us to investigate whether certain human activities, commonly performed by an instructor in a classroom, can be modeled and predicted. However, the dataset was not collected in a classroom setting. We believe since the data source, which are the sensors in a smartwatch with the instructor, remains the same as that of the data here, therefore, the findings in this paper will remain valid in classroom setting.

5 CONCLUSION AND FUTURE WORKS

In this paper, we have presented both traditional and neural network based models to classify 4 different activities (sit, walk, computer work, stand) in multi-label learning settings with class imbalance problem. The proposed system shows promising results on activity recognition using smartwatch accelerometer data. As a part of our contributions, we have presented different metrics scores (AUC, BA, F1) for classifying 4 activities. We have also tried to solve 3 open problems described in [25], which are time series modeling, multi-task modeling and feature learning.

A lot of future work can be done based on this research. More activities can be included. The proposed models can be tested on actual smartwatch data collected from instructors inside classroom. Currently, only accelerometer data is used. Gyro sensor and microphone audio data can be collected from smartwatch and sensor fusion can be done to classify more activities of instructors like talking, writing etc.

ACKNOWLEDGMENTS

This work is supported in part by the National Science Foundation under grant DUE-1525208. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Z. Abdallah, M. Gaber, B. Srinivasan, and S. Krishnaswamy. 2018. Activity Recognition with Evolving Data Streams: A Review. *ACM Computing Surveys (CSUR)* 51, 4 (2018), 71.

[2] P. Casale, O. Pujol, and P. Radeva. 2011. Human Activity Recognition from Accelerometer Data using a Wearable Device. In *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 289–296.

[3] P. Domingos. 1999. Metacost: A General Method for Making Classifiers Cost-sensitive. In *KDD*, Vol. 99. 155–164.

[4] P. Donnelly, N. Blanchard, B. Samei, A. Olney, X. Sun, B. Ward, S. Kelly, M. Nystran, and S. D'Mello. 2016. Automatic Teacher Modeling from Live Classroom Audio. In *Proceedings of the 2016 conference on user modeling adaptation and personalization*. ACM, 45–53.

[5] C. Droge. 1998. Know your Customer: New Approaches to Understanding Customer Value and Satisfaction. *Journal of the Academy of Marketing Science* 26, 4 (1998), 351.

[6] S. Elrahman and A. Abraham. 2013. A Review of Class Imbalance Problem. *Journal of Network and Innovative Computing* 1, 2013 (2013), 332–340.

[7] J. Hanley and B. McNeil. 1982. The Meaning and use of the Area Under a Receiver Operating Characteristic (ROC) Curve. *Radiology* 143, 1 (1982), 29–36.

[8] S. Hochreiter and J. Schmidhuber. 1997. Long Short-Term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[9] S. Ke, H. Thuc, Y. Lee, J. Hwang, J.Yoo, and K. Choi. 2013. A Review on Video-based Human Activity Recognition. *computers* 2, 2 (2013), 88–131.

[10] D. Kleinbaum, K. Dietz, M. Gail, and M. Klein. 2002. *Logistic Regression*. Springer.

[11] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, et al. 2006. Handling Imbalanced Datasets: A review. *GESTS International Transactions on Computer Science and Engineering* 30, 1 (2006), 25–36.

[12] B. Krawczyk. 2016. Learning from Imbalanced Data: Open Challenges and Future Directions. *Progress in Artificial Intelligence* 5, 4 (2016), 221–232.

[13] M. Lawton and E. Brody. 1969. Assessment of Older People: Self-maintaining and Instrumental Activities of Daily Living. *The gerontologist* 9, 3_Part_1 (1969), 179–186.

[14] A. Liaw, M. Wiener, et al. 2002. Classification and Regression by RandomForest. *R news* 2, 3 (2002), 18–22.

[15] F. Murtaza, M. Yousaf, and S. Velastin. 2017. PMHI: Proposals from Motion History Images for Temporal Segmentation of Long Uncut Videos. *IEEE Signal Processing Letters* 25, 2 (2017), 179–183.

[16] N. Nida, M. Yousaf, A. Irtaza, and S. Velastin. 2019. Bag of Deep Features for Instructor Activity Recognition in Lecture Room. In *International Conference on Multimedia Modeling*. Springer, 481–492.

[17] W. Niu, J. Long, D. Han, and Y. Wang. 2004. Human Activity Detection and Recognition for Video Surveillance. In *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, Vol. 1. IEEE, 719–722.

[18] M. Panwar, S. Dyuthi, K. Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, and G. Naik. 2017. CNN based Approach for Activity Recognition using a Wrist-Worn Accelerometer. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2438–2441.

[19] S. Ramasamy Ramamurthy and N. Roy. 2018. Recent Trends in Machine Learning for Human Activity Recognition—A Survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, 4 (2018), e1254.

[20] S. Safavian and D. Landgrebe. 1991. A Survey of Decision Tree Classifier Methodology. *IEEE transactions on systems, man, and cybernetics* 21, 3 (1991), 660–674.

[21] M. Satyanarayanan et al. 2001. Pervasive Computing: Vision and Challenges. *IEEE Personal communications* 8, 4 (2001), 10–17.

[22] M. Shoaib, S. Bosch, O. Incel, H. Scholten, and P. Havinga. 2016. Complex Human Activity Recognition using Smartphone and Wrist-Worn Motion Sensors. *Sensors* 16, 4 (2016), 426.

[23] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas. 2008. Multi-label Classification of Music into Emotions. In *ISMIR*, Vol. 8. 325–330.

[24] G. Tsoumakas and I. Katakis. 2007. Multi-label Classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)* 3, 3 (2007), 1–13.

[25] Y. Vaizman, K. Ellis, and G. Lanckriet. 2017. Recognizing Detailed Human Context in the Wild from Smartphones and Smartwatches. *IEEE Pervasive Computing* 16, 4 (October 2017), 62–74. https://doi.org/10.1109/MPRV.2017.3971131

[26] Y. Vaizman, N. Weibel, and G. Lanckriet. 2018. Context Recognition In-the-Wild: Unified Model for Multi-Modal Sensors and Multi-Label Classification. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article Article 168 (Jan. 2018), 22 pages. https://doi.org/10.1145/3161192

[27] N. Veliyath, P. De, A. Allen, C. Hodges, and A. Mitra. 2019. Modeling Students' Attention in the Classroom using Eyetrackers. In *Proceedings of the 2019 ACM Southeast Conference*. ACM, 2–9.

[28] S. Wang. 2003. Artificial Neural Network. In *Interdisciplinary computing in java programming*. Springer, 81–100.